

MAXI MANAGER
A DATA BASE MANAGEMENT SYSTEM

Written By
EXADOR, INC.
265 Hollyberry Court
Roswell, Georgia 30076

TRS-80 Model I Appendix
Version A.3 (48K)
April 1981

Published and Distributed By
Adventure International
A Division of Scott Adams Inc.
P.O. Box 3435
Longwood, Florida 32750
(305) 862-6917

FOREWORD

I would like to take this opportunity to congratulate you on your decision to purchase the Version A.3 upgrade to the MAXI MANAGER System. I believe that you will find this program to be one of the most useful and frequently used additions to your software library.

MAXI MANAGER has been under development for over two years and has undergone several transformations along the way. The program, first introduced in October, 1980 under the name of MAXI MICRO-MANAGER, has been steadily growing in popularity. The most recent change involves both the title and the software.

When Radio Shack introduced the TRS-80 Model III, the existing file structure was unable to take advantage of the double density features available in the new machine. In addition, the introduction of new and more sophisticated Disk Operating Systems capable of supporting mixed combinations of disk drives (5 1/4" or 8" or both) and varying numbers of tracks per drive (35, 40, 77, or 80) necessitated the redesign of MAXI MANAGER.

The introduction of MAXI MANAGER System Version A.3 for the TRS-80 Models I and III marks the beginning of a new era in TRS-80 data management systems. Only changes to the original manual are contained in this A.3 appendix.

Owners of earlier versions of the program will be pleased to know that all existing data files are upward compatible following the use of the UPGRADE3 Extension Program conversion supplied on the UTILITY (B) disk. The UPGRADE3 Extension Program converts data files created with earlier versions of the MAXI MANAGER to the new A.3 version format. Data files undergoing change through the UPGRADE3 process include both MASTER/MMS and MATH/MMS. Note that only the MMS data disks containing the POINTER/MMS files must be converted (usually drive #0 unless created with the earlier NEWDOS/80 version); all other disks comprising a multiple disk data file require NO conversion.

We at Exador, Inc. are continually striving to improve both the features and performance of MAXI MANAGER and rely on your suggestions and comments. Please feel free to drop us a line. Many of the features included in the A.3 version came directly from users of the program. In this regard, MAXI MANAGER will continue to grow. In the offering are a multi-level SORT, an extract program, a reconfiguration program, increased number of fields and field lengths, user designed video displays, and a non-destructive cursor.

We are also striving to support all Disk Operating Systems possible. Write for details if yours is not covered by the DOSPATCH/BAS program described on page 4 of this manual.

With your support, MAXI MANAGER will continue to evolve as time and needs dictate.

Dale A. Kubler
President - Exador, Inc.

MAXI MANAGER

TABLE OF CONTENTS

Loading TRSDOS.....	1
BACKUP Procedures.....	1
Optional Operating Systems.....	2
Initializing DATA disks.....	5
Define New File.....	6
Multiple Key Search.....	9
Sort Records.....	11
Utility Programs.....	.
UPGRADE1.....	11
UPGRADE2.....	11
UPGRADE3.....	11
INFOXFER.....	12
AIDSXFER.....	12
DOCUFILE.....	14
File Structures.....	18
MASTER/MMS.....	19
POINTER/MMS.....	22
RECORDS/MMS.....	23
MATH/MMS.....	24
MMSBASIC.....	26
Program Descriptions.....	.
PROGRAM (A) Disk.....	31
UTILITY (B) Disk.....	32
ERROR Codes.....	34
Technical Support.....	34

LOADING TRSDOS:

TRSDOS is the property of the Tandy Corporation and as such cannot be distributed with this software package.

For your convenience, a TRSDOS transfer program is included on each of the MAXI MANAGER disks; the transfer program automatically executes the first time the disk is placed in drive #0 and the reset key is depressed.

You must initialize both the 'MMS PROGRAM (A) Disk' and the 'MMS UTILITY (B) Disk' with TRSDOS 2.3. This is accomplished by loading the MMS disks (PROGRAM and UTILITY) into drive # 0 and depressing the reset key. The transfer utility 'DOSXFER' will then prompt you as necessary. If the TRSDOS 2.3 transfer does not function properly on the first try, the DOSXFER/CMD program (stored on the UTILITY Disk) should be executed directly from DOS using a 2 disk system.

If you desire, NEWDOS, NEWDOS/80, and DBLDOS are also supported but CANNOT be transferred to the MMS disks using this procedure. Refer to the Optional Operating Systems portion of this manual (pages 2-4) for additional instructions.

BACKUP PROCEDURES:

The BACKUP function is executed by performing the following steps:

1. LOAD a copy of TRSDOS 2.3 containing BACKUP into drive # 0.
2. Depress the 'reset' button causing TRSDOS to load and display the DOS READY prompt.
3. Type BACKUP and depress the ENTER key.
4. After the BACKUP utility is loaded, remove the TRSDOS disk and replace with one of the MAXI MANAGER disks.
5. In response to the prompt 'SOURCE Drive Number?', ENTER the drive number containing the MMS disk. In response to the prompt 'DESTINATION Drive Number?', ENTER the drive number that you will be BACKing UP to. For single drive systems, the SOURCE and DDESTINATION drive numbers should be zero (0). Single drive systems will be required to swap SOURCE and DESTINATION disks at various stages of the BACKUP procedure.
6. You will next be prompted to ENTER the 'BACKUP DATE (MM/DD/YY)?'. Type in the current date. As soon as you depress ENTER the BACKUP will begin.

***** NOTE *****

The MAXI MANAGER disks cannot be backed up (copied) until TRSDOS 2.3 has been loaded onto each disk. Failure to load TRSDOS 2.3 before backing up the disks will result in a 'Destination Disk Read Error' message.

OPTIONAL OPERATING SYSTEMS:

The MAXI MANAGER System (MMS) is now compatible with Apparat's NEWDOS, NEWDOS/80, and Percom's DBLDOS operating systems. The program disks supplied with MMS are formatted for the standard 35 track version of TRSDOS 2.3 (which must be transferred to the MMS PROGRAM (A) disk and the MMS UTILITY (B) disk via the auto loading DOSXFER/CMD program stored on each disk prior to initialization). Failure to transfer TRSDOS 2.3 prior to BACKUP will cause the invalid error message of 'Destination Disk Read Error' to appear. This problem can be overcome however by using the NEWDOS, NEWDOS/80, or DBLDOS 'COPY' command.

CREATING A MINIMUM SYSTEM DISK:

The instructions that follow assume a general knowledge of DOS System and Disk BASIC commands.

In order to use the NEWDOS, NEWDOS/80, or DBLDOS operating systems, a reduced size NEWDOS/DBLDOS disk must be created. This is accomplished by KILLING files not required by MAXI MANAGER and is necessary in order to make room for the MMS program files. The command:

DIR (I,S)

will list all files currently stored on a disk and should be used to determine which files must be KILLED. The following files MUST reside on the SYSTEM disk:

NEWDOS	NEWDOS/80	DBLDOS
-----	-----	-----
SYS0/SYS	SYS0/SYS	SYS0/SYS
SYS1/SYS	SYS1/SYS	SYS1/SYS
SYS2/SYS	SYS2/SYS	SYS2/SYS
SYS3/SYS	SYS3/SYS	SYS3/SYS
SYS4/SYS	SYS7/SYS	SYS4/SYS
SYS6/SYS	SYS10/SYS	SYS6/SYS
BOOT/SYS	BOOT/SYS	SYS7/SYS
DIR/SYS	DIR/SYS	DIR/SYS
COPY/CMD (NOTE 1)	SYS6/SYS (NOTE 2)	COPY/CMD (NOTE 1)
BASIC/CMD (NOTE 1)	BASIC/CMD (NOTE 1)	BASIC/CMD (NOTE 1)

NOTE 1:

Drive #0 data disks do not require these files.

NOTE 2:

This SYSTEM file contains the NEWDOS/80 COPY command and should be KILLED if sufficient space is not available on your disk for all of the MAXI MANAGER programs. This file is not required on Drive #0 data disks.

NOTE 3:

MAXI MANAGER will operate with NEWDOS/80 modified for use with DoubleZAP 2. DO NOT ENABLE *ADR*.

You should now make a BACKUP copy of this disk as you will need one for the UTILITY (B) and one for the PROGRAM (A). The next step is to COPY the required MMS program files from the TRSDOS 2.3 disk (supplied with purchase) to the NEWDOS/DBLDOS disk. In the instructions that follow, <ENTER> means to depress the ENTER key.

A. Two Drive System

1. Power-up the system.
2. Place the newly created NEWDOS/DBLDOS disk in drive #0 and depress the RESET button.
3. Place the TRSDOS 2.3 PROGRAM (A) disk in drive #1.
4. Type the following:

```
VERIFY (ON) <ENTER>
COPY $MMS:1 TO MMS:0 <ENTER>
COPY $INITIAL/MMS:1 TO INITIAL/MMS:0 <ENTER>
COPY $PRINT/MMS:1 TO PRINT/MMS:0 <ENTER>
COPY $SORT/MMS:1 TO SORT/MMS:0 <ENTER>
COPY $LOADER/CMD:1 TO LOADER/CMD:0 <ENTER>
COPY $DATAPLUS/CMD:1 TO DATAPLUS/CMD:0 <ENTER>
COPY $PRINTM/CMD:1 TO PRINTM/CMD:0 <ENTER>
COPY $QSORT/CMD:1 TO QSORT/CMD:0 <ENTER>
AUTO LOADER <ENTER>
```

5. Remove the NEWDOS/DBLDOS disk from drive #0. Mount the second copy of the NEWDOS/DBLDOS disk into drive #0.
6. Remove the MMS PROGRAM (A) disk from drive #1.
7. Place the TRSDOS 2.3 MMS UTILITY (B) disk in drive #1.
8. Type the following:

```
COPY $REPORT/DF:1 TO REPORT/DF:0 <ENTER>
COPY $LABEL1/DF:1 TO LABEL1/DF:0 <ENTER>
COPY $LABEL2/DF:1 TO LABEL2/DF:0 <ENTER>
COPY $MASTER/MMS.MMS:1 TO MASTER/MMS.MMS:0 <ENTER>
COPY $POINTER1/MMS.MMS:1 TO POINTER1/MMS.MMS:0 <ENTER>
COPY $POINTER2/MMS.MMS:1 TO POINTER2/MMS.MMS:0 <ENTER>
COPY $RECORDS/MMS.MMS:1 TO RECORDS/MMS.MMS:0 <ENTER>
COPY $MATH/MMS:1 TO MATH/MMS:0 <ENTER>
COPY $UPGRADE1:1 TO UPGRADE1:0 <ENTER>
COPY $UPGRADE2:1 TO UPGRADE2:0 <ENTER>
COPY $UPGRADE3:1 TO UPGRADE3:0 <ENTER>
COPY $INFOXFER:1 TO INFOXFER:0 <ENTER>
COPY $AIDXFER:1 TO AIDXFER:0 <ENTER>
COPY $DOCUFILE:1 TO DOCUFILE:0 <ENTER>
COPY $DOSPATCH/BAS:1 TO DOSPATCH/BAS:0 <ENTER>
COPY $DOSPATCH/DAT:1 TO DOSPATCH/DAT:0 <ENTER>
```

B. Single Drive System

1. Follow steps one and two as described for a two drive system.
2. Follow step four except change the format of the COPY command

```
from:          COPY $MMS:1 TO MMS:0
to:            COPY :0 $MMS TO MMS
```

Place appropriate disks in drive #0 when requested.

'SOURCE' = The TRSDOS 2.3 version of the MMS PROGRAM (A) disk.

'SYSTEM' = The NEWDOS/DBLDOS disk.

'DESTINATION' = The NEWDOS/DBLDOS disk.

3. Remove the NEWDOS/DBLDOS disk from drive #0.
Load the second copy of the reduced size disk into drive #0.
4. Follow step eight except use the same COPY format as described in step two above.

PATCHING THE PROGRAM FILES:

Once you have created your MMS PROGRAM (A) and UTILITY (B) disks, you're ready to begin the second phase of the operating system conversion.

1. Place the newly created NEWDOS/DBLDOS UTILITY (B) disk in drive #0 and type:

BASIC RUN"DOSPATCH/BAS".

2. The screen will clear and a MENU will appear giving you options for patching one of seven (7) operating systems. The program has taken into account the VTOS 4.0, LDOS, and DOSPLUS operating systems however; it is not known at this time whether or not MAXI MANAGER is compatible with these three DOS's. We are awaiting delivery on all three for testing. Write to EXADOR, Inc. for further info (please include a self addressed stamped envelope).
3. Remove the UTILITY (B) disk from drive #0 when instructed to do so by the program.
4. Mount the NEWDOS/DBLDOS PROGRAM (A) disk on drive #0. Depress <ENTER> when ready.
5. The DOSPATCH/BAS program will now complete the patching of the MMS program files for NEWDOS/DBLDOS compatibility.

INITIALIZING DATA DISKS:

The MAXI MANAGER System is designed to provide you with the maximum number of records for any configuration that you construct. One method used to increase the number of available records is to use drive #0 for data storage. MMS allows you to store data on drive # 0 for either large data files or for single disk systems. This is accomplished by using a 'minimum configuration' disk in drive # 0. If this option is used, MMS will automatically prompt you to swap the PROGRAM and DATA disks as necessary. If you plan to store data on drive # 0, you must create a SYSTEM DATA DISK with the smallest possible amount of System programming on it to be used as the drive # 0 data disk.

CREATING A MINIMUM SYSTEM DISK:

The instructions that follow assume a general knowledge of DOS System and Disk BASIC commands.

The command:

DIR (I,S)

will list all files currently stored on a disk and should be used to determine which files must be KILLED. The following files MUST reside on the 'minimum system' disk:

TRSDOS

SYS0/SYS
SYS1/SYS
SYS2/SYS
SYS3/SYS
SYS4/SYS
SYS6/SYS
BOOT/SYS
DIR/SYS

NEWDOS

SYS0/SYS
SYS1/SYS
SYS2/SYS
SYS3/SYS
SYS4/SYS
SYS6/SYS
BOOT/SYS
DIR/SYS

NEWDOS/80

SYS0/SYS
SYS1/SYS
SYS2/SYS
SYS3/SYS
SYS7/SYS
SYS10/SYS
BOOT/SYS
DIR/SYS

DBLDOS

SYS0/SYS
SYS1/SYS
SYS2/SYS
SYS3/SYS
SYS4/SYS
SYS6/SYS
SYS7/SYS
DIR/SYS

DEFINE NEW FILE:

The DEFINE NEW FILE function will probably be the first Master Menu command executed. For this reason, a comprehensive and complete understanding of the MAXI MANAGER System DEFINITIONS is in order. If you haven't already done so, READ the DEFINITIONS section of your original manual before proceeding any further.

Once you are familiar with MAXI MANAGER's vocabulary, you must define the system under which MAXI MANAGER is to operate. This is accomplished through the use of a 'DATA' statement implanted in the INITIAL/MMS program that resides on the PROGRAM (A) disk. The DATA statement in line 222 contains a total of twelve (12) independent variables that describe the configuration of your computer system. The format of the DATA statement is as follows:

```
222 DATA X(0),Y(0),Z(0),X(1),Y(1),Z(1),X(2),Y(2),Z(2),X(3),Y(3),Z(3)
```

where X(n) equals the number of FREE grans available on disk drive #n,
Y(n) equals the number of BYTES per granule on disk drive #n, and
Z(n) equals the number of BYTES per sector on disk drive #n.

On a standard Model I system, the value of X(n) for a 'minimum system' DATA Disk (35 tracks) is equal to 57, the value for a FORMATED DATA Disk (35 track) is 67. The value of Y(n) is equal to 1280 and the value of Z(n) is equal to 256.

Through the proper setting of these twelve variables, any type of system configuration can be supported.

Field labels, field types, field lengths, field equations, and number of Key Files are the five major components ENTERED by the user during file definition. This information, along with the number of data disks to be allocated to the file, will determine the size of your data base. A program to calculate the maximum number of records your file will hold is available as a Utility Program for those who wish to maximize the data base size by using the optimum set of initialization parameters. Note that the file is always created in its maximum size. An MMS Utility Program, to be sold separately, will enable you to reconfigure an existing MAXI MANAGER file to include a larger number of disk drives and/or Key Files).

Selection of the Define New File function will cause MMS to load the Initialization program from the PROGRAM (A) disk and will prompt you to:

ENTER the description of field # X:

where X represents the number of the lowest unassigned field. MMS supports a maximum of twenty (20) fields with labels of nineteen (19) characters each. Following entry of the field label, MMS will display:

ENTER the length of field # X:

MMS will accept any one of seven responses:

Field Type	Valid Responses
Alphanumeric Field	1 thru 40
Date Field	D-
Equation Field (Numeric)	E1 thru E40
Equation Field (Rounded Numeric)	E1 thru E17
Extended Date Field	D+
Numeric Field	-1 thru -40
Rounded Numeric Field	-1 thru -17

Respond accordingly. For example, if a data field will only contain numeric information such as a zip code, assign its field length using a numeric field. In the case of a zip code, the response would be -5. This will cause the MAXI MANAGER System to reject any alpha characters.

Whenever a numeric, rounded numeric, or equation field is requested, MAXI MANAGER will prompt you to:

ENTER the number of required decimal places (0-??)

where the '??' are replaced by the highest number allowed. When using rounded numeric fields, don't forget to allow for the decimal point and the sign (+ or -) of the numeric value when determining the length of the field. If the field chosen is an equation field, read the paragraphs describing equation fields appearing at the end of this section.

Successful entry of the initialization information causes MMS to format the display and generate the following sub-commands:

'C' - Change Field X	'N' - Input Field X+1
'D' - Delete Field X	'@' - End Initialization

Entry of sub-command 'C' will re-request the field initialization information for field X, 'D' will delete field X, and 'N' will continue the initialization process. When all fields have been assigned, ENTER the '@' symbol to 'End Initialization. The screen will clear and display the following:

Number of disk drives (1-4)

Your answer determines the number of disk drives that are to be allocated to the data base about to be created. Following your selection here, MMS will ask you for the:

Drive number of the first DATA Disk (0-3)

Respond accordingly. Your answer to this question is used by MMS to determine which disk will be the first for this file. If you wish to leave the PROGRAM (A) disk in drive #0 then answer this question with a number greater than zero. Remember, if drive #0 is used for data storage, the 'minimum system' as described on page 16 is required to be on the drive #0 disk. Next, MMS will ask for the:

Drive number of Key File Storage Disk (?-?)

Your input here is limited to a number greater than or equal to the number of the first data disk provided in the previous question and less than or equal to three. Following your answer to this question, you will be asked for the:

Number of Key Files (1-5)

Respond to this prompt with the smallest number that you can get by with. Remember, a Key File is used for every unique SORT sequence you wish to keep 'on line' at any given moment without having to reSORT. For further information regarding the use of Key Files, refer to appropriate sections of the DEFINITIONS portion of this manual as well as the SORT and Key File Maintenance commands. One Key File is all that is required for the majority of applications.

At this point, the program begins calculating maximum file size using recursive techniques. This may take from three seconds to two minutes; do not despair if the system appears to freeze temporarily. When completed, the system will display the following message:

The Initialization Process is now complete.

Currently defined fields and disk drives allow for the retention of XXXXX records.

Insert a Blank 'SYSTEM' disk into Drive # 0

The 'SYSTEM' Disk must contain SYSTEM files only !!!!!

Insert a Blank 'FORMATTED' disk into all other drives.

NOTE:

The MMS PROGRAM Disk MUST be removed at this time if drive #0 is going to be used for data storage. In its place should be the 'minimum system' disk created earlier. If all DATA Disks are not available as specified, ENTER a 'SHIFT @' or an '@' to return to the Master Menu. This is the only means of escape available to you at this point short of rebooting the system.

The MAXI MANAGER System will now take control of your computer for an extended period of time depending upon configuration of the system and data base. It will now initialize all DATA Disks building the required files to their maximum size.

That's all there is to it. Upon completion of Initialization, MMS will prompt you to:

Load the
PROGRAM Disk into Drive # 0

and return you to the Master Menu.

MULTIPLE KEY SEARCH:

The Multiple Key Search functions supported are:

1. Instring
2. Not Instring
3. Equal
4. Not Equal
5. Greater/Less Than

Additionally, MAXI MANAGER supports:

1. Logical AND
2. Logical OR

between all fields as well as within any single field. In addition, a Masked Search function is available.

The Multiple Key Search feature will display the current 'Key #' that you are defining in the upper right hand portion of the video display. The program will display the field labels (which may be toggled from page one to page two with the UP ARROW and DOWN ARROW Special Function Keys) along with the following prompt:

'@' - Cancel All Keys

ENTER the field number to be searched:

'*' - Return to Menu

To return to the previous Menu, ENTER an '*' symbol. If during the definition phase of the Multiple Key Search you desire to start over again, ENTER an '@' symbol otherwise ENTER the field number you wish to define as a Search Key. Following input of the Key field, MAXI MANAGER displays:

1 - Instring, 2 - Not Instring, 3 - Equal, 4 - Not Equal

ENTER the relational search operator: 5 - Greater/Less Than

You must now define the type of search to perform. Next you will be prompted as follows:

'1' - Logical AND

Specify search type: '2' - Logical OR

At this time, you should identify the 'Logical AND/OR' field operators. These field operators are for search filters within a field. Basically, a 'Logical AND' tells MAXI MANAGER to find a match for ALL Search Keys defined as having the same field number whereas a 'Logical OR' instructs MMS to find a match for at least one (but not necessarily ALL) Search Keys having the same field number.

NOTE 1:

Once a field number has been defined as a Search Key, a match must be found for at least one Search Key for that field regardless of the 'Logical AND/OR' relationships defined unless you instruct MMS to perform a 'Logical OR' between fields. What this means is that you cannot say "Find all records having a CITY equal to New York or a STATE equal to New York". If you do, only those records with a CITY equal to New York AND a STATE equal to New York will be retrieved unless you instruct MMS to perform a 'Logical OR' between fields. A record with CITY and STATE fields of Poughkeepsie and New York respectively will be bypassed because the CITY Search Key did not equal New York when performing a 'Logical AND' between fields; however, Poughkeepsie, New York would be retrieved if a 'Logical OR' between fields was requested since at least one of the two fields contained the Search Key of New York.

NOTE 2:

If an 'Instring' search of all fields is desired, ENTER a field number equal to zero. Fields having a field length smaller than the number of characters defined in the Search Key will be ignored.

Finally, MAXI MANAGER will ask you to:

ENTER the search key:

If the search operator was defined as 'Greater/Less Than', MMS will request:

Beginning search key:

Ending search key:

Respond accordingly. Since field number ten is a rounded numeric field defined with a field length equal to 7.2 which equates to a field seven characters long with two decimal places, the search key must contain the leading blanks required since all rounded numeric fields are right justified when stored. The 'Beginning search key' should be defined as a single blank space and the 'Ending search key' MUST be defined as three (3) blanks followed by 1.00 making a total of seven right justified characters.

Having defined the Search Key, MAXI MANAGER will prompt you to respond to the following:

'N' - Next Field
'B' - Begin Searching

'C' - Cancel this Field
'X' - Cancel Field & Begin Searching

If you are finished defining the Multiple Key Search, type 'B' and ENTER to begin the actual search. On the other hand, if you wish to cancel the Search Key that you have just defined AND begin the actual search, type 'X' and ENTER. If you wish to cancel the Search Key you have just defined, type 'C' and ENTER; or if you wish to define another Search Key, type 'N' and ENTER. In either case you will be returned to the

'@' - Cancel All Keys

'*' - Return to Menu

ENTER the field number to be searched:

prompt at which point you should take the appropriate action.

SORT RECORDS:

NOTE 1:

When SORTing a portion of a file, the current Key File number must be the same as the Key File number that will receive the SORT results. If not, you will be returned to the Master Menu.

NOTE 2:

The next release of MAXI MANAGER (Version A.4) will contain a three (3) level SORT option. Registered owners will be notified of its availability by mail.

UTILITY PROGRAMS:

UTILITY Sub-Command Function 1:

This function is used to RUN additional software applications programs as they are introduced.

Supplied with the MAXI MANAGER are three Extension Programs which allow you to convert Data Files created by The Bottom Shelf, Inc.'s DATA MANAGER* program or Dale Kubler's DATA MANAGER II# program to MAXI MANAGER compatible Data Files. The first, entitled UPGRADE1 will perform Step 1 of the conversion process (FOR DATA MANAGER* ONLY); UPGRADE2, the second program performs Step 2 of the conversion and should only be RUN after UPGRADE1 has been executed. UPGRADE2 should be RUN on both DATA MANAGER* and DATA MANAGER II# files. The third program, UPGRADE3, should be run by both DATA MANAGER*, DATA MANAGER II#, and all earlier versions of MICRO-MANAGER\$ (A.0 - A.2) and MAXI MICRO-MANAGER\$ (A.0 - A.2) files and should be the last program run. For MICRO-MANAGER\$ and MAXI MICRO-MANAGER\$ files, it should be the ONLY conversion program run.

Prior to UPGRADING either DATA MANAGER* or DATA MANAGER II# DATA Disks, the filenames of the DATA Disk files must be renamed using the DOS 'RENAME' command. The OLD and NEW filenames to be used in the RENAME command are:

OLD Filename	NEW Filename
MASTER/DMS.password	MASTER/MMS.password
POINTER1/DMS.password	POINTER1/MMS.password
POINTER2/DMS.password	POINTER2/MMS.password
POINTER3/DMS.password	POINTER3/MMS.password
POINTER4/DMS.password	POINTER4/MMS.password
POINTER5/DMS.password	POINTER5/MMS.password
RECORDS/DMS.password	RECORDS/MMS.password

- * Registered Trademark of The Bottom Shelf, Inc.
- # Registered Trademark of Dale A. Kubler
- \$ Registered Trademark of EXADOR, Inc.

As you can see, the change involves the file extension which must now be 'MMS' rather than 'DMS'. All files on all DATA Disks MUST be renamed. To RENAME a file, mount the DATA Disk(s) into their proper drives and 'reset' the computer. The display should clear and read:

DOS READY

You should now begin renaming your DATA Disk(s) ENTERING the RENAME command for each DMS data file in the following format:

RENAME old-filename TO new-filename

Remember, to view the directory of a DATA Disk, ENTER the DOS command:

DIR :n

where n corresponds to the DATA Disk directory you wish to view.

A fourth Extension Program entitled INFOXFER allows you to transfer and/or merge Data Files created by The Bottom Shelf Inc.'s INFO SYSTEM* program into MAXI MANAGER Data Files. Both cassette tape and disk INFO SYSTEM* files are supported by the INFOXFER program. INFO SYSTEM* files to be merged into a MMS Data File MUST contain the same field labels and field lengths as those in the MMS Data File. If this is not the case, the merge will abort. The INFOXFER Extension Program will create a MAXI MANAGER Data File, if desired, that can then be used for all subsequent merges. One last note, the MMS Data File must contain sufficient spare records to be able to merge the entire INFO SYSTEM* file; if this is not the case, break the INFO SYSTEM* file into smaller segments (i.e. smaller data files).

A fifth Extension Program entitled AIDSXFER allows you to transfer and/or merge data files created with Metatechnologies Corporation, Inc.'s AIDS-II# or AIDS-III# programs into the MAXI MANAGER. Prior to using the program, you must first create an AIDS 'descriptor' and an AIDS 'data' file having a file name of eight or less characters. The AIDSXFER program is run in the same manner as INFOXFER and retains the same restrictions regarding field lengths and field names.

The sixth and last Extension Program stored on the UTILITY (B) disk is one entitled DOCUFILE. In that it is the Extension Program used to create Document Files (see the PRINT FILE Function pages 38-59 of your original manual), a separate section of this manual (pages 14-17) is devoted to it's use.

* Registered Trademark of The Bottom Shelf, Inc.

Registered Trademark of Metatechnologies Corporation, Inc.

When instructed to:

ENTER the name of the Extension Program:

type the name of the desired Extension Program. You will then be prompted to insert the disk containing the Extension Program into Drive # 0. Remember, the Extension Programs are stored on the UTILITY (B) Disk supplied with MAXI MANAGER. Each Extension Program is self-explanatory by way of video display prompts.

UTILITY Sub-Command Function 3:

The function of this sub-command is to calculate the total number of records the MAXI MANAGER System will support. The user is prompted to ENTER the following information:

Number of disk drives (1-4) =
Drive number of first DATA Disk (0-3) =
Drive number of Key File storage disk (0-3) =
Number of Key Files (1-5) =
Number of characters per record (1-800) =

MMS will then calculate the 'Total number of records allowed for this configuration'. The function is intended to be used to determine the maximum record size, minimum number of disk drives, and/or number of Key Files.

Example:

Suppose you are going to create a file consisting of ten characters per record with two Key Files. On a system using four drives with 256 bytes per sector, 1280 bytes per granule, 57 granules FREE on drive #0, 67 granules FREE on drives #1-3, and forcing the Key Files to be stored on drive #0, MMS will support a total of 16512 records. Calculating the maximum file size using only three drives yields the same maximum file size of 16512 records. It is therefore evident that additional file capacity CANNOT be obtained on a four drive system and, as such, a three drive system will suffice. This is due to the fact that all Key Files are stored on the 'minimum system' DATA Disk and that no additional POINTER file space is available. Taking this problem one step further, if only two drives are used (drives #0-1), MMS will support a total of 10700 records. As can be seen thru the use of this utility, the third drive would add an additional 5812 records.

The utility can also be used to determine the maximum record size assuming that a data base has to be created that could handle X number of records. In this case, through trial and error, the optimum system configuration containing X records can be obtained. As stated earlier in these instructions, the record size calculation routines are iterative and without the use of a computer, could take an hour or more to manually perform.

DOCUFILE:

An Extension Program entitled DOCUFILE has been developed to take the pain out of building Document Files. The DOCUFILE program resides on the UTILITY (B) disk of MMS and is executed via the Extension Program command described on pages 40-42.

Before you build any Document Files, you should be aware of a few major points.

1. Document Files are independent of the MAXI MANAGER data files. Be certain that sufficient space exists on the disk that will hold the Document File before you begin building a new file.
2. Several Document Files can reside on the same disk. YOU assign Document File names to each Document File you create. The DOCUFILE program will assign the default file extension of '/DF' to each file unless instructed by you to do otherwise.
3. Don't panic. Document Files are nothing more than a blueprint for use by the PRINT FILE Function of MMS. The Document File merely contains the necessary information required to instruct MMS as to how you want your reports, labels, and letters printed.
4. DOCUFILE can be used to build NEW files only. It cannot be used to EDIT or make changes to an existing Document File. Such activity is best handled by a traditional word processing type program capable of reading, changing, and writing ASCII files.

The first question asked by DOCUFILE is:

Do you wish to read Field Labels from an MMS data file (Y/N)?

If your answer is 'Y', DOCUFILE will instruct you to insert a DATA disk from the file you wish to work with into one of the spare disk drives. As always, this disk drive can be drive #0 for single drive systems provided that the disk in question contains the 'minimum system'. Following this step, you will be instructed to:

ENTER the name of your Document File:

Do not include a disk drive extension number. You will be asked for the drive number to store the Document File later. Your response is limited to eight characters. As previously stated, the default file extension of '/DF' will be assigned to the file unless otherwise instructed. The next question will be:

Which disk drive do you wish to store this file on (0-3)

Respond accordingly. Having completed the housekeeping chores, you will be presented with the following Menu:

Document File Creation Menu

- 0 - Exit
- 1 - Page Definition
- 2 - Margins
- 3 - Page Numbering
- 4 - Printer Controls
- 5 - Print Drivers
- 6 - Report Generator
- 7 - Label Generator
- 8 - Text

Most of this should be self-explanatory. The recommended order for defining a Document File is to first select the Print Driver routines. Then, routines other than Row/Column entries for reports and labels should follow. It is best to group like commands together. Each Menu will now be presented. Please note when viewing each Menu that items enclosed within brackets <> are the current settings for each function for the Document File you are building. Refer to pages 51-59 of your original manual for detailed descriptions of each PFF command option.

Page Definition Menu

- 0 - Exit
- 1 - Characters per Line <60>
- 2 - Lines per Page <66>
- 3 - Space Between Lines <1>
- 4 - Comment Line

Margin Menu

- 0 - Exit
- 1 - Top Margin <6>
- 2 - Left Margin <10>
- 3 - Bottom Margin <6>
- 4 - Left and Right Indentation <0>
- 5 - Right Justification <No>
- 6 - Comment Line

Page Numbering Menu

- 0 - Exit
- 1 - Print Page Numbers <No>
- 2 - Define Page Number Format
- 3 - Stop at End of Page <No>
- 4 - Comment Line

Page Numbering Menu

- 0 - Exit
- 1 - Page Label Left Margin <1>
- 2 - Page Label Header <Yes>
- 3 - Page Label Footer <No>
- 4 - Page Label Description <Page>
- 5 - Comment Line

Printer Control Command Menu

- 0 - Exit
- 1 - Printer Control Codes
- 2 - Null Characters <0>
- 3 - Form Feed
- 4 - Stop at End of Page <No>
- 5 - Space Between Lines <1>
- 6 - Comment Line

Print Driver Selection Menu

- 0 - Exit
- 1 - Parallel Port - Centronics <Yes>
- 2 - RS-232-C Serial Port <No>
- 3 - TRS-232 Serial Cassette Port <No>
- 4 - Comment Line

Report Generator

- 0 - Exit
- 1 - Define Page
- 2 - Define Margins
- 3 - Define Report Title
- 4 - Establish New Row
- 5 - Define Column Entries
- 6 - Define Printer Control Codes
- 7 - Establish Print Drivers
- 8 - Comment Line

Label Generator

- 0 - Exit
- 1 - Define Page
- 2 - Define Margins
- 3 - Establish New Row
- 4 - Define Column Entries
- 5 - Define Printer Control Codes
- 6 - Establish Print Drivers
- 7 - Comment Line

Text Input Menu

- 0 - Exit
- 1 - Enter Text
- 2 - Review MMS Field Labels
- 3 - Comment Line

When you have finished defining your Document File, you will be returned to the Utility Program Menu from which you first entered the DOCUFILE program.

FILE STRUCTURES:

In this section, the MAXI MANAGER file structures will be revealed. Initially, the structures may appear to be complex but as you will soon learn, the file structures are straight forward and easy to comprehend once you understand the concepts involved. An MMS data base is comprised of four main files:

1. MASTER/MMS.password:PD
2. RECORDS/MMS.password:PD
3. POINTER1/MMS.password:PD
4. MATH/MMS:PD

where 'password' is the password you assigned to the file and 'PD' is the drive number you specified for Key File storage during initialization. The POINTER files are actually the Key Files and always reside on drive # PD. The number that follows the word POINTER equates to the Key File number that the POINTER file represents. For example:

Key File Number	Equivalent POINTER File
1	POINTER1/MMS.password:PD
2	POINTER2/MMS.password:PD
5	POINTER5/MMS.password:PD

The MASTER file contains the initialization parameters (i.e. field labels, date field flags, maximum number of available records, etc.) as well as the data disk number, data file serial number, etc. The MASTER file appears on all data disks but only the MASTER file on drive # PD is updated as records are added or deleted.

The MATH/MMS file resides on drive # PD and contains the translated equation field algorithms used by MAXI MANAGER.

Last, but not least, is the RECORDS file. This is the file that contains the actual data entries of an MMS data base. RECORDS files appear on each data disk and account for the largest portion of the data disks. We will now discuss each file in detail.

MASTER/MMS:

The MASTER file variables used in MAXI MANAGER are defined as follows:

Variable Name	Variable Function
A(n).....	An integer value equal to the maximum number of records stored on drive # n
AF(n).....	An integer variable equal to the absolute value of the field length
CF(n).....	An integer variable flag equal to one if the field is an Equation field and equal to zero otherwise
D2.....	An integer variable equal to the total number of fields
DN.....	The integer Data Disk ID used to ensure that Data Disks are mounted in the proper drives
DF(n).....	An integer Date field flag equal to zero if not a Date field, equal to one if a Date field, and equal to two if an Extended Date field
DP(n).....	An integer value equal to the number of desired decimal places for field n
DR\$(n).....	A string variable equal in length to the field length of the field it represents
F(0).....	An integer value equal to the sum of the field lengths
F(n).....	An integer value equal to the field length. Negative values signify Numeric, Date, Rounded Numeric, or Equation fields
F(D2+1).....	An integer value equal to the length of the largest field
F\$(n).....	A string variable equal to the field label
FD.....	An integer variable equal to the disk drive number (0-3) containing the first data disk.
IW.....	An integer variable equal to the number of spare LINKED records. This value is decremented until equal to zero. Each time a record is deleted, this variable is incremented by one (1).
KF(0).....	An integer value equal to the number of the active Key File

KF(n).....An integer value equal to the field number assigned to Key File n

KF(6).....An integer value equal to the first field number to be used in Abbreviated Video displays

KF(7).....An integer flag equal to zero if the current Key File is to be read in ascending order and equal to one if the Key File is to be from from in descending order.

ND.....An integer value equal to the total number of disks in the data base less one

NE(1).....An integer value equal to NE(2)*NE(3). This variable is used by the MMSBASIC file accessing routines

NE(2).....An integer value equal to the number of subrecord blocks used by the MMSBASIC filing routines

NE(3).....An integer value equal to the length of a subrecord block

NF\$(n).....A string value equal in length to NE(3). The highest value of n equals NE(2)

NK.....An integer value equal to the maximum number of Key Files

PD.....An integer variable equal to the disk drive number containing the MATH, POINTER, and active MASTER files.

R.....An integer value equal to the number of active records in the data base

R(0).....An integer serial number unique to a data file

R(1).....An integer serial number unique to a data file

R(2).....An integer serial number unique to a data file

SR.....An integer value equal to the number of the next spare record location

V1.....An integer variable equal to the maximum number of records allocated to the data base

The following program listing provides the necessary routines to read the MASTER file variables:

```
10 CLEAR 1000:DEFINT A-Z
20 DIM AF(20),CF(20),CF$(20),DF(20),DF$(20),DP(20),DP$(20),DR$(20)
25 DIM F(21),F$(20),FS$(20)
30 OPEN"R",1,"MASTER/MMS.password:PD"
```

```
40 FIELD#1,2 AS ND$,2 AS DD$, 2 AS NK$,2 AS V1$,2 AS FD$,2 AS PD$,  
    2 AS IW$,2 AS D2$,2 AS R$,2 AS SR$,2 AS VR$(0),2 AS VR$(1),  
    2 AS VR$(2),2 AS VR$(3),8 AS IV$,2 AS R$(0),2 AS R$(1),2 AS R$(2)  
50 FOR I=0 TO 7  
60 FIELD#1,42+I*2 AS X1$,2 AS KF$(I)  
70 NEXT I  
75 GET#1,1  
80 ND=CVI(ND$): DD=CVI(DD$): NK=CVI(NK$): V1=CVI(V1$): D2=CVI(D2$)  
90 R=CVI(R$): SR=CVI(SR$): IW=CVI(IW$): FOR I=0 TO 2  
100 A(I)=CVI(VR$(I))  
100 R(I)=CVI(R$(I)): NEXT I  
110 A(3)=CVI(VR$(3))  
120 FOR I=0 TO 7  
130 KF(I)=CVI(KF$(I))  
140 NEXT I  
150 IF KF(0)=0 THEN KF(0)=1 "Must have a valid Key File number"  
160 F(0)=0: F(21)=0  
170 FOR I=1 TO D2  
180 FIELD#1,(I-1)*2 AS X1$,2 AS FS$(I)  
190 FIELD#1,128+(I-1)*2 AS X2$,2 AS DF$(I)  
200 FIELD#1,168+(I-1)*2 AS X3$,2 AS CF$(I)  
210 FIELD#1,208+(I-1)*2 AS X4$,2 AS DP$(I)  
220 NEXT I  
230 FIELD#1,D2*2 AS X5$,2 AS NE$(1),2 AS NE$(2),2 AS NE$(3)  
240 GET#1,2  
250 FOR I=1 TO 3  
260 NE(I)=CVI(NE$(I))  
270 NEXT I  
340 FOR I=1 TO D2  
350 F(I)=CVI(FS$(I))  
360 AF(I)=ABS(F(I))  
370 F(0)=F(0)+AF(I)  
380 DF(I)=CVI(DF$(I))  
390 CF(I)=CVI(CF$(I))  
400 DP(I)=CVI(DP$(I))  
410 IF AF(I)>F(21) THEN F(21)=AF(I)  
420 NEXT I  
430 PP=0  
440 FOR I=1 TO D2  
450 IF I/13=INT(I/13) THEN PP=1  
460 FIELD#1,(I-PP*12-1)*20 AS X1$,20 AS FS$(I)  
470 IF PP=1 THEN GET#1,5 ELSE GET#1,3  
480 F$(I)=FS$(I)  
490 NEXT I  
500 CLOSE#1
```

POINTER/MMS:

The POINTER files are made up of a series of two byte compressed integer string variables. The files are equal in length to the highest active record number. Each variable contains an integer pointer or key to the actual storage location of the random record it represents. For example, the data associated with record number seven is accessed by reading the seventh two byte string variable contained in the POINTER file and converting the string to an integer. Then, with the help of MMSBASIC, the data record associated with the integer variable is retrieved from the RECORDS file.

When a file is SORTed, the pointers representing the data records are shuffled and written to the proper POINTER file in ascending order.

Several Key Files are possible within an MMS data base. Whenever the current Key File is changed, MMS merely changes the POINTER file number it reads from to access the actual data.

A Key File can be accessed in descending order by merely accessing the pointers from last to first.

The following program provides the necessary routines to read the POINTER files:

```

10 CLEAR 500: DEFINIT A-Z
15 DEFFNTT=ABS(T-AD*(R+1)) 'R=Highest active record number
20 IN$="POINTER/MMS,password:PD": RL=2: PI=5: EC=0
25 RW=0: T1=1: PR$=STRING$(2,32)
30 NAME MPEN(IN$,PI,RL,EC)
45 AD=0 'Zero (0) read Key File in ascending order and
      'one (1) reads in descending order
60 T=1 'T equals the record number desired
65 T=FNTT
80 NAME DALE(PI,T,T1,RW,PR$,EC)
90 PR=CVI(PR$) 'PR equals the integer pointer value

```

RECORDS/MMS:

To retrieve records from a MAXI MANAGER data base, the following steps must be performed:

1. Read MASTER/MMS variables
2. Open (MPEN) the RECORDS/MMS file(s)
3. Initialize the destination block and field variables

Then for each record desired:

4. Retrieve record block(s)
5. CUT block(s) into field elements

And finally, before exiting the program:

6. Close (SHUT) the RECORDS/MMS file(s)

Once the MASTER/MMS file variables have been read, the RECORDS/MMS file(s) must be opened.

```
1000 EC=0 'Error Code initialization
1005 'FD equals the disk drive number containing
      the first DATA disk (MASTER file)
1010 'ND equals the number of DATA Disks minus one (MASTER file)
1020 'NE(3) equals the record block length (MASTER file)
1030 FOR I=0 TO ND
1040 FM$="RECORDS/MMS."+ "password"+" ":"+CHR$(48+I)
1050 FM=I+1 'File number
1060 NAME MPEN(FM$,FM,NE(3),EC)
1070 NEXT I
```

Next the block and field destination variables must be initialized. Failure to initialize the destination string variables will result in a fatal error when calling the MMSBASIC commands, FILE and CUT.

```
2000 'AF(D2) equals an array containing the absolute value field lengths
2010 'D2 equals the number of fields
2020 'NE(2) equals the number of record blocks
2030 'NE(3) equals the record block length
2040 DIM NF$(NE(2)),DR$(D2)
2050 FOR I=1 TO NE(2)
2060 NF$(I)=STRING$(NE(3),32) 'Initialize block
2070 NEXT I
2080 FOR I=1 TO D2
2090 DR$(I)=STRING$(AF(I),32) 'Initialize fields
2100 NEXT I
```

Page 24

Repeated as often as necessary, the following code retrieves the data from the RECORDS file(s). The DR\$() array will contain the actual data record where DR\$(1) corresponds to the contents of field number one, DR\$(2) corresponds to the contents of field number two, etc.

```

3000 'A(0) equals the MASTER file variable containing end of disk info
3005 'PD equals the disk drive number where the Key Files,
      MATH/MMS file, and the updated MASTER file reside.
3010 EC=0 'Error Code initialization
3020 'NE(2) equals the number of record blocks to retrieve
3040 RW=0 'indicates a record is to be read from the data file -
      a one (1) indicates data is to be written to the data file
3050 T=1 'record number to be accessed
3060 NAME FILE(FD,ND,T,A(0),NE(2),NF$(1),RW,EC)
3070 NAME CUT(NE(2),D2,NF$(1),DR$(1))

```

NOTE: ALL VARIABLES USED IN THE MMSBASIC COMMANDS MUST BE DEFINED AS INTEGER VARIABLES UNLESS OTHERWISE DEFINED. IF THE DEFINIT FUNCTION SUPPLIED IN THE MASTER/MMS FILE READING CODE IS ADHERED TO, NO PROBLEMS WILL ARISE.

MATH/MMS:

The MATH/MMS file stored on the System DATA Disk contains the translated equation functions. If equation fields were not used in a data file, the MATH/MMS file will be empty. The ASCII file, MATH/MMS, contains a series of program statements which are automatically MERGED into the program at RUN time. The statement numbers range from 500 - 690 with the equation statement for field number one beginning with line 500. Statements are incremented by a factor of 10.

If you desire to change the equations for a field, you must LOAD the MATH/MMS file as a BASIC program, make the necessary changes, and SAVE the changed file with the ASCII option. When viewing a translated equation, you will see that all F#nn statements ENTERed during initialization have been replaced by the statement:

DR\$(nn)

Be sure to use the same format when changing equation fields. For example, field #9 is defined as an Equation field on the sample data base contained on the UTILITY (B) disk provided with MMS. The formula ENTERed during initialization was:

F#7*(1+F#8/100)

The translated formula appearing in the MATH/MMS file stored on the UTILITY (B) disk is:

580 DR\$(9)=DR\$(7)*(1+DR\$(8)/100):LSETDR\$(9)=STR\$(DR\$(9))

IDLE LINKED LIST:

The missing link to understanding the MMS file structures is the Idle Link List (no pun intended). The strategy employed by an Idle Link List is rather simple:

EACH SPARE RECORD POINTS TO THE NEXT SPARE RECORD

The last record of the file contains an 'end of file' mark rather than the next spare record. In the MAXI MANAGER System, a zero (0) is used to signify the 'end of file'. An Idle Link List may be viewed as follows:

Record No.	File
10	: 9* :-----
9	-----: 8* :<-----
8	: ----->: 7* :-----
7	-----: 6* :<-----
6	: ----->: 5* :-----
5	-----: 4* :<-----
4	: ----->: 3* :-----
3	-----: 2* :<-----
2	: ----->: 1* :-----
1	-----: 0* :<-----
	End of File

Note that all spare record pointers are followed by an asterisk. To ADD a new record to the file, the variable IW is checked to see if it is greater than zero (0). If it is equal to zero, the next record written in the data base is equal to the number of the record itself. That is to say that if IW=0 and the highest active record in the file is equal to 27, then the next record to be written will record number 28.

If IW is greater than zero, the value of IW is decremented by one (1). Then the 'next available spare record' pointer variable is retrieved from the MASTER file. The record number contained in this variable is then read in an effort to obtain the NEW 'next available spare record'. The MASTER file variable is updated and the record data is written into the file.

Page 26

NEW MMSBASIC COMMANDS:

CUT ***** NEW FORMAT *****

Format: NAME CUT (NB%, D2%, NF\$(1), DR\$(1))

DR\$(1) is the first element of the destination array for an MMS record split into field components, NF\$(1) is a string variable array. Each element must be equal in length to the record blocks. NB% is an integer variable containing the number of record blocks, D2% is an integer variable equal to the number of fields contained in the data file, and AF\$(1) is the first element of an integer array containing the absolute value of the field lengths.

Purpose: To split the MMS record blocks into MMS record field components.

Remarks: The following conventions apply:

- 0 < LEN(NF\$(1)) <= 255
- 0 < LEN(DR\$(1)) <= 40
- 1 <= D2% <= 20
- 1 <= AF\$(1) <= 40
- 0 < NB% <= 4

Example 1:

```
10 AF$(1)=20: AF$(2)=20: AF$(3)=40
15 D2%=3 'Number of fields
20 FOR I=1 TO D2%
22 DR$(I)=STRING$(AF$(I),32)
24 NEXT I
30 NB%=1
32 NF$(I)=STRING$(80,50)
50 NAME CUT (NB%, D2%, NF$(1), DR$(1))
```

FILE ***** NEW FORMAT *****

Format: NAME FILE(FD%,ND%,RN%,A%(0),NB%,NF\$(1),RW%,EC%)

where A%(0) is an integer variable array whose elements equal the number of records stored on each disk, FD% is an integer variable equal to the drive number of the first data disk, ND% is an integer variable equal to the total number of disks in the data file minus one, RN% is an integer variable equal to the MMS record number desired (i.e. translated POINTER file record number), NB% is an integer equal to the number of record blocks, RW% is an integer variable equal to zero (0) if data is to be retrieved and a one (1) if data is to be stored, NF\$(1) is a string array equal in length to the size of the record block (i.e. the record length in which the data files were MPENed, and EC% is an integer variable ERROR code.

Purpose: To read/write a data record from a MAXI MANAGER data file. Record blocking, sector spanning, and disk spanning is automatically taken care of as a function of the command.

Remarks: The following conventions apply:

```
LEN(NF$(1)) <= 255
1 <= NB% <= 4
0 <= RN% <= 32767
0 <= RW% <= 1
0 <= FD% <= 3
0 <= ND% <= 3
```

Example 1:

```
10 A%(0)=500
15 A%(1)=600
17 FD%=0
18 ND%=1
20 RN% = 200 'Record number to access
25 REM ***** RL% equals the file's record length
26 REM ***** NB% equals the number of sub-blocks
28 FOR I=1 TO NB%
30 NF$(I)=STRING$(RL%,32)
33 NEXT I
35 RW%=0 'Retrieve flag
40 EC% = 0
50 NAME FILE(FD%,ND%,RN%,A%(0),NB%,NF$(1),RW%,EC%)
```

NOTE: The ERROR Code EC% is greater than zero if an ERROR has occurred. ERROR trapping is left up to the programmer and is not linked to Disk BASIC's ON ERROR GOTO functions. The ERROR codes returned are TRSDOS ERROR codes rather than BASIC or Disk BASIC ERROR codes.

GLUE ***** NEW FORMAT *****

Format: NAME GLUE (D2%,NB%,DR\$(1),NF\$(1))

DR\$(1) is the first element of the source array of a MMS record. The GLUE command is used to pack the individual field components into the filing array NF\$(n). NF\$(n) is a string variable array. Each element must be equal in length to the record blocks, AF\$(1) is the first element of an integer array containing the absolute value of the field lengths, NB% is an integer variable containing the number of record blocks, and D2% is an integer variable equal to the number of fields contained in the data file.

Purpose: To pack the MMS record field components into MMS record blocks.

Remarks: The following conventions apply:

0 < LEN(NF\$(1)) <= 255
0 < LEN(DR\$(1)) <=40
1 <= D2% <= 20
1 <= AF\$(1) <= 40
0 < NB% <= 4

Example 1: 10 AF\$(1)=20: AF\$(2)=20: AF\$(3)=40
15 D2%=3 Number of fields
20 FOR I=1 TO D2%
22 DR\$(I)=STRING\$(AF\$(I),50+I)
24 NEXT I
30 NB%=1
32 NF\$(I)=STRING\$(80,32)
50 NAME GLUE (D2%,NB%,DR\$(1),NF\$(1))

LOWER ***** NEW COMMAND *****

Format: NAME LOWER (IN\$,X%)

where IN\$ is a string expression or string array and X% is an integer variable equal to one (1) if IN\$ is a string variable and equal to the number of elements of an array to be converted to upper case if IN\$ is equal to an array. The conversion will start with the first element passed to MMSBASIC in the NAME LOWER command.

Purpose: To convert lower case string variables to upper case string variables.

Remarks: If variable IN\$ is an array, the value of X% must be less than or equal to the Dimension of IN\$(). If the Dimension of IN\$() equals ten (10), and the IN\$ argument given in the LOWER command is IN\$(4), then the value of X% must be greater than zero and less than or equal to 10-4 or six (6).

Example 1: 10 B\$ = "All good men must come to the aid of their country"
20 X%=1
30 NAME LOWER(B\$,X%)
40 'The contents of B\$ are now:
ALL GOOD MEN MUST COME TO THE AID OF THEIR COUNTRY

Example 2: 10 DIM A\$(15)
20 X%=14
30 NAME LOWER(A\$(2),X%)
40 'Array elements 2-15 will be converted to upper case

Page 30

NKEY ***** CORRECTED FORMAT *****

Format: NAME NKEY (IN\$,IN%)

where IN% is an integer expression and IN\$ is a string expression.

Purpose: To provide an input mechanism which displays and limits the length of the input field. An ideal way of preventing input errors.

Remarks: The integer variable IN% must be set equal to the length of the desired input string prior to calling NKEY. The following conventions apply:
 IN% > 0 for alphanumeric values
 IN% < 0 for numeric values
 IN% = 0 will cause an unrecoverable ERROR

The string variable IN\$ must be set equal to a string whose length is equal to the absolute value of IN%. Upon detection of the ENTER key, control is return to BASIC with IN\$ equal to the input response. If the value of IN\$ is numeric, standard use of the VAL(IN\$) BASIC command can be used to obtain the numeric value for assignment to another variable. A call to NKEY automatically executes the BREAK command.

Example 1: 10 REM *** ALPHANUMERIC INPUT
 20 IN% = 12
 30 IN\$ = STRING\$(ABS(IN%),32)
 40 NAME NKEY(IN\$,IN%)

Example 2: 10 REM *** NUMERIC INPUT
 20 IN% = -5
 30 IN\$ = STRING\$(ABS(IN%),32)
 40 NAME NKEY(IN\$,IN%)

NOTE: Control is returned to BASIC if the first character keyed is either the UP ARROW, DOWN ARROW, CLEAR, or BREAK. The ASCII codes returned in IN\$ are as follows:

KEY	ASCII Decimal Code
BREAK	1
CLEAR	31
DOWN ARROW	10
UP ARROW	91

In addition, the Screen Print function is activated if the first and only character keyed is a PERIOD followed by the ENTER key. Following the Screen Print, control is returned back to the NKEY function with the PERIOD removed.

PROGRAM DESCRIPTIONS:

The MAXI MANAGER System consists of two disks labeled 'A' and 'B'. The 'A' disk contains the MAXI MANAGER programs and will be the disk most often used. The 'B' disk, referred throughout this manual as the 'UTILITY Disk contains a small sample data base as well as several utility programs.

PROGRAM (A) DISK:

<u>Program Name</u>	<u>Format</u>	<u>Function</u>
LOADER/CMD	Object Code	Contains the initial loader display as well as a link to DATAPLUS/CMD
MMSBASIC/CMD	Object Code	Deletes unused files from the PROGRAM Disk leaving only modules necessary for MMSBASIC routines. Also links other CMD files together for use as MMSBASIC
DATAPLUS/CMD	Object Code	The heart of MMSBASIC. Always resident in memory except when SORTING or in the PRINT FILE Function
QSORT/CMD	Object Code	The machine language portion of the SORT module.
PRINTM/CMD	Object Code	The machine language portion of the PRINT FILE Function.
MMS	BASIC	The main program.
SORT/MMS	BASIC	The BASIC portion of the SORT/MERGE program.
INITIAL/MMS	BASIC	The initialization and utility programs module.
PRINT/MMS	BASIC	The BASIC portion of the PRINT FILE Function.

UTILITY (B) DISK:

<u>Program Name</u>	<u>Format</u>	<u>Function</u>
MASTER/MMS.MMS	Data	The MASTER file of sample data base.
POINTER1/MMS.MMS	Data	Key File #1 for the sample data base.
POINTER2/MMS.MMS	Data	Key File #2 for the sample data base.
RECORDS/MMS.MMS	Data	The data storage file for the sample data base.
MATH/MMS	ASCII	The file containing the equation field formulae of the sample data base.
REPORT/DF	ASCII	A Report Document File for the sample data base.
LABEL1/DF	ASCII	A Label Document File for the sample data base.
LABEL2/DF	ASCII	A Label Document File for the sample data base.
DOSPATCH/BAS	BASIC	The patching program for the NEWDOS, NEWDOS/80, and DBLDOS operating systems.
DOSPATCH/DAT	Data	The data file used by the DOSPATCH/BAS program.
DOSXFER/CMD	Object Code	The TRSDOS 2.3 transfer program.
DOCUFILE	BASIC	The DOCUFILE extension program used to build Document Files.
INFOXFER	BASIC	The extension program used to transfer data from The Bottom Shelf's INFO SYSTEM to MAXI MANAGER.
AIDSXFER	BASIC	The extension program used to transfer data from Metatechnology's AIDS II or AIDS III programs to MAXI MANAGER.

UTILITY (B) DISK (continued):

Program Name	Format	Function
UPGRADE1	BASIC	The extension program used to upgrade data files created with The Bottom Shelf's DATA MANAGER program to MAXI MANAGER
UPGRADE2	BASIC	The extension program used to upgrade data files created with The Bottom Shelf's DATA MANAGER II or Dale Kubler's DATA MANAGER II program to MAXI MANAGER
UPGRADE3	BASIC	The extension program used to upgrade data files created with Exador, Inc.'s MAXI MICRO-MANAGER program (TRSDOS, NEWDOS, DBLDOS, or NEWDOS/80 versions) to MAXI MANAGER Version A.3

ERROR CODES:

The MAXI MANAGER System has been extensively error trapped and is believed to be 'bug free' provided that the instructions contained in the manual are adhered to. In the unlikely event that an error does occur, the following message will appear:

Error Code - XXX-YY

The first three digits (XXX) of the Error Code represent the BASIC program line number of the error. When writing in regard to a program question or problem relating to a suspected error, refer to the Error Code. The last two digits (YY) represent a Level II and/or Disk Basic error code.

Once an ERROR has occurred, it is almost always non-recoverable. The program should be exited, through function number eight (Close Files and End) if possible.

Try running the program again. If the ERROR continues, write to EXADOR, Inc. for assistance. When writing, please include copies of the DATA disks, the passwords used, the Disk Operating System used, and a description of how the problem can be reproduced. Any additional information will be helpful. If the ERROR occurred as a result of user or equipment problems, you will be billed (following a written or verbal estimate) for the restoration time involved.

TECHNICAL SUPPORT:

While every effort has been made to test this software package for errors and incompatibilities, there may be occasions where we missed a potential 'bug'. Your help is solicited if any are found. Prior to contacting Exador, Inc., please reread the documentation, and verify that the problem is indeed a software error.

WRITE, DO NOT CALL. Few, if any, problems can be solved over the telephone. Send a detailed written explanation of the circumstances that led to the problem and how it can be duplicated. Be sure to include copies of all data disks used for the application (as well as passwords if used) so that we may duplicate the error on our end. The contents of all DATA Disks will be destroyed if requested; the confidentiality of your data is guaranteed.

Our address is:

EXADOR, Inc.
265 Hollyberry Court
Roswell, Georgia 30076

We do take telephone calls on Wednesday evenings between the hours of 7:30 and 9:30 pm Eastern time. We DO NOT take calls outside of these hours.

***** MAXI MANAGER Version A.3 *****

Features include:

1. Upper/Lower case search (doesn't care how ENTERed)
2. Numeric Masked Search
3. Granule allocation of file size (like NEWDOS/80 version)
4. Program can remain in drive #0
5. NO IDLE LINKED LIST during initialization - create a new file of 10000 records in 2 minutes.
6. Not Instring Search.
7. Logical 'AND' between fields as well as within fields.
8. Increased speed for equation fields.
9. DOCUFILE utility to automatically build Document Files
10. AIDSXFER utility to transfer Metatechnologies' AIDS II and AIDS III data files.